Listing of the Claims

A complete listing of the pending claims, indicating the status of each claim, follows.

1. (Withdrawn) A method for generating a file object identifier comprising the steps: (a) allocating memory for said identifier; (b) storing in said allocated memory the value of the disk volume holding the file object; (c) storing in said allocated memory the value of the disk block holding the file object; and (d) storing in said allocated memory the value of the offset within said disk block holding the file object, said offset computed in multi-byte increments.

2. (Withdrawn) The method of claim 1 wherein said file object is one of a file, a directory, and a symbolic link.

3. (Withdrawn) The method of claim 1 wherein said memory allocated for said identifier is 32 bits.

4. (Withdrawn) The method of claim 1 wherein the value of the disk volume holding the file object is stored in 4 bits of said allocated memory.

5. (Withdrawn) The method of claim 1 wherein the value of the disk block holding the file object is stored in 23 bits of said allocated memory.

6. (Withdrawn) The method of claim 1 wherein the value of the offset within said disk block holding the file object is stored in 5 bits of said allocated memory.

7. (Withdrawn) The method of claim 1 wherein the value of the multi-byte offset increment within said disk block holding the file object is at least 128 bytes.

8. (Withdrawn) The method of claim 1 wherein said file object identifier is a POSIX file serial number.

9. (Cancelled).

10. (Amended). The method of ~~claim 9~~ claim 29 wherein said file object is one of a file, a directory, and a symbolic link.

11. (Amended). The method of ~~claim 9~~ claim 29 wherein said second bit size is less than said first bit size.

12. (Amended). The method of ~~claim 9~~ claim 29 wherein said first file object identifier comprises a disk volume value, a disk block value and a block offset value.

13. (Amended). The method of ~~claim 9~~ claim 29 wherein said at least one file system characteristic comprises limiting the number of disks available in any logical volume to a 4 bit value.

14. (Amended). The method of ~~claim 9~~ claim 29 wherein ~~said at least one file system characteristic comprises limiting~~ the second file object identifier limits address granularity within a disk block to at least 32 bytes.

15. (Amended). The method of ~~claim 9~~ claim 29 wherein ~~said at least one file system characteristic comprises limiting~~ the second file object identifier limits file object lengths to at least 128 bytes.

16. (Amended). The method of ~~claim 9~~ claim 29 wherein ~~said~~ the second file object identifier is a POSIX file serial number.

17-24. (Cancelled).

25. (Withdrawn) A fault-tolerant computer having a proprietary operating system and support for standards-compliant file operations comprising: two central processing units (CPUs), operating synchronously; two memory modules, each associated with one of said CPUs; an operating system, providing operating system functionality and comprising a standards-compliant interface and a proprietary interface; and an application program, invoking said standards-compliant interface.

26. (Withdrawn) The fault-tolerant computer of claim 22 wherein said proprietary operating system is Stratus Virtual Operating System (VOS).

27. (Withdrawn) The fault-tolerant computer of claim 22 wherein said standards-compliant file operations are POSIX file operations.

28. (Withdrawn) The fault-tolerant computer of claim 22 wherein said standards-compliant interface is a POSIX interface.

29. (Original) A method for mapping a first file object identifier having a first bit size to a second file object identifier having a second bit size comprising the steps:

(a) receiving said first file object identifier associated with a file object;

(b) extracting a disk block value and a disk volume value from said first file object identifier;

(c) locating a file object in a location on a disk specified by said extracted disk block value and said extracted disk volume value;

(d) computing a temporary file object identifier for said located file object;

(e) iterating step (d) for file objects in said specified location on the disk until the temporary file object identifier matches said first file object identifier;

(f) computing a second file object identifier for said file object with said temporary file object identifier matching said first file object identifier; and

(g) providing said second file object identifier.

30. (Original) The method of claim 29 wherein said first file object identifier is a POSIX file serial number.